

# Multivariate Time Series Forecasting with Deep Learning

Kao Panboonyuen, Ph.D.

Contact: [panboonyuen.kao@gmail.com](mailto:panboonyuen.kao@gmail.com)

Code: [https://github.com/kaopanboonyuen/GISTDA\\_TRAINING\\_2023](https://github.com/kaopanboonyuen/GISTDA_TRAINING_2023)

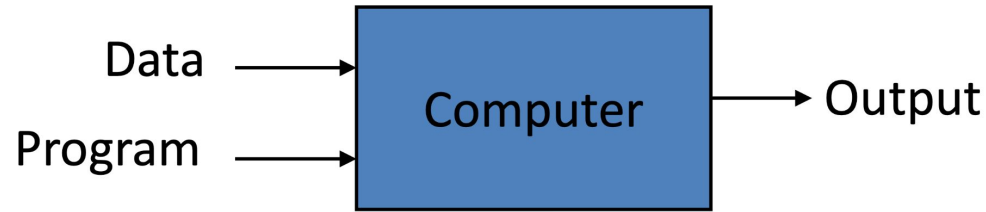
# Outlines

1. Introduction to Machine Learning/Deep Learning
2. Basic Pandas
3. Regression
4. SARIMAX and Exercise with GISTDA Ocean Current Data
5. illustrated Guide to LSTM and GRU

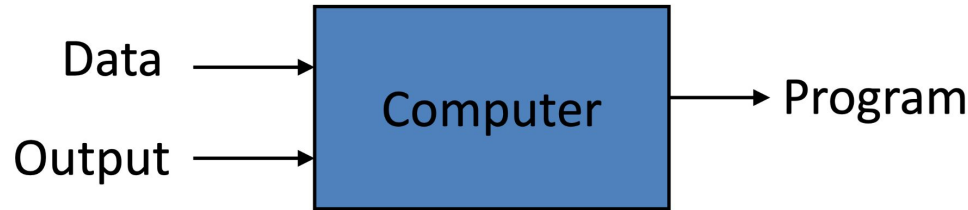
Code: [https://github.com/kaopanboonyuen/GISTDA\\_TRAINING\\_2023](https://github.com/kaopanboonyuen/GISTDA_TRAINING_2023)

# Introduction to Machine Learning/Deep Learning

## Traditional Programming



## Machine Learning



# Introduction to Machine Learning/Deep Learning (Cont.)

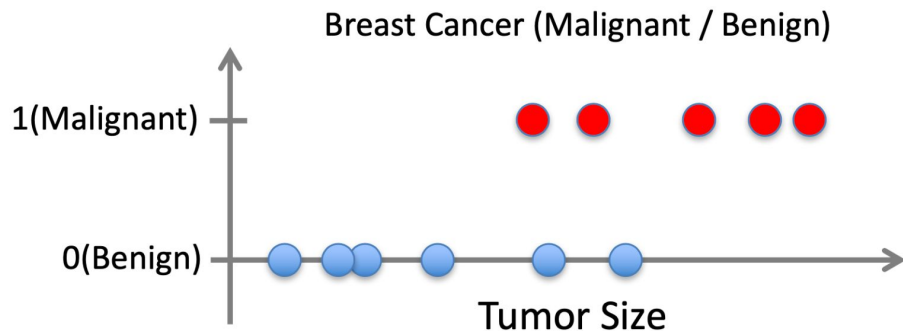
## Types of Learning

- Supervised (inductive) learning
  - Given: training data + desired outputs (labels)
- Unsupervised learning
  - Given: training data (without desired outputs)
- Semi-supervised learning
  - Given: training data + a few desired outputs
- Reinforcement learning
  - Rewards from sequence of actions

# Introduction to Machine Learning/Deep Learning (Cont.)

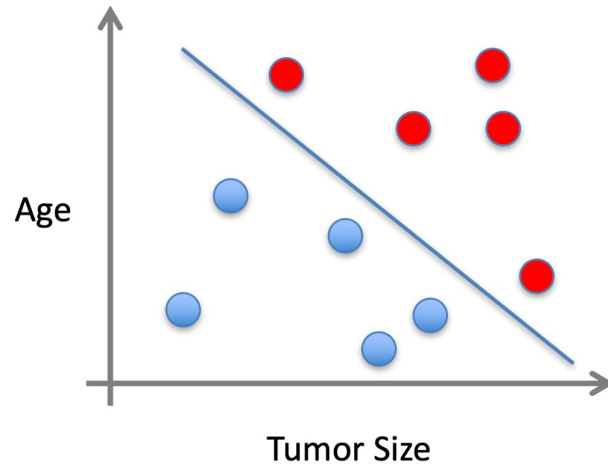
## Supervised Learning: Classification

- Given  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function  $f(x)$  to predict  $y$  given  $x$ 
  - $y$  is categorical == classification



# Introduction to Machine Learning/Deep Learning (Cont.)

- $x$  can be multi-dimensional
  - Each dimension corresponds to an attribute

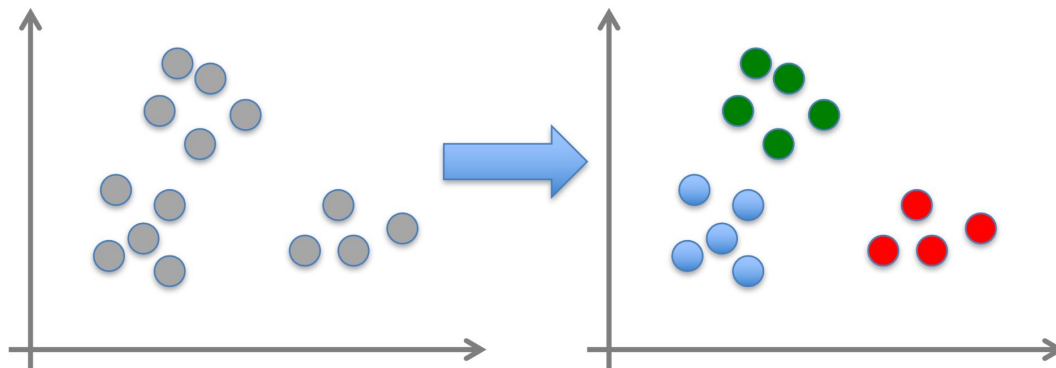


- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape
- ...

# Introduction to Machine Learning/Deep Learning (Cont.)

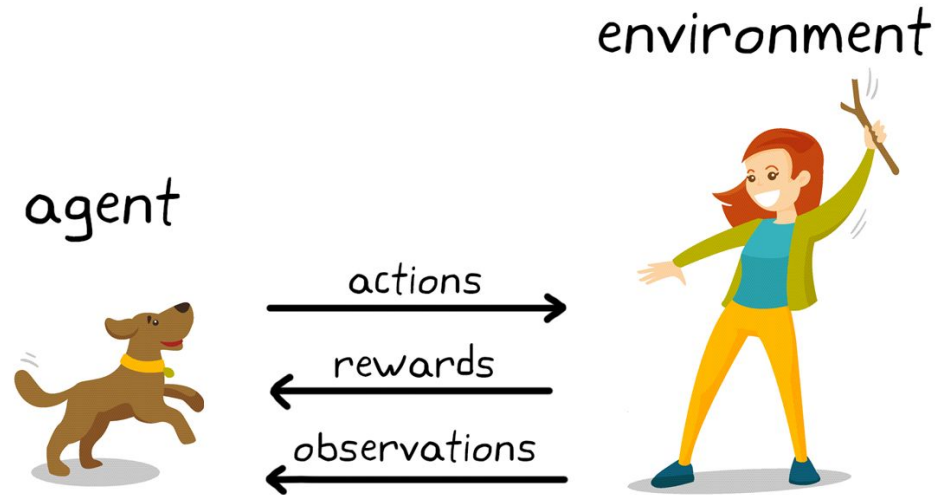
## Unsupervised Learning

- Given  $x_1, x_2, \dots, x_n$  (without labels)
- Output hidden structure behind the  $x$ 's
  - E.g., clustering



# Introduction to Machine Learning/Deep Learning (Cont.)

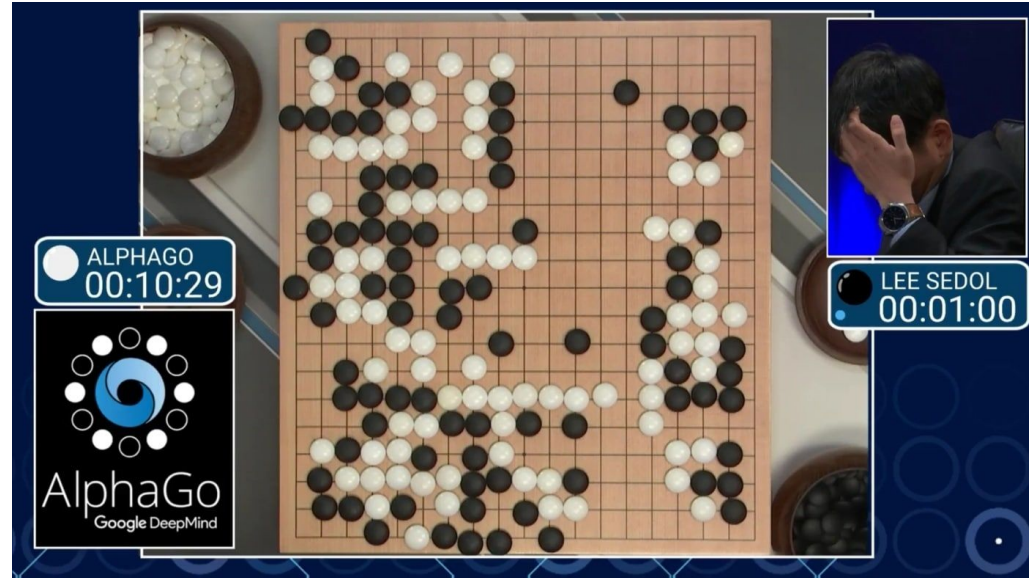
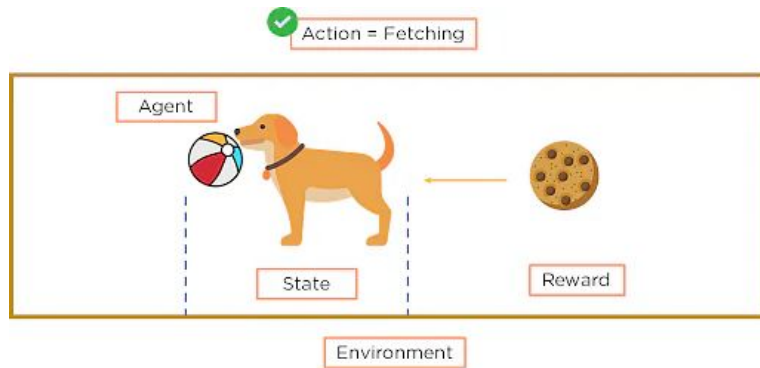
## Reinforcement Learning





# Introduction to Machine Learning/Deep Learning (Cont.)

## Reinforcement Learning



# Introduction to Machine Learning/Deep Learning (Cont.)

## Reinforcement Learning



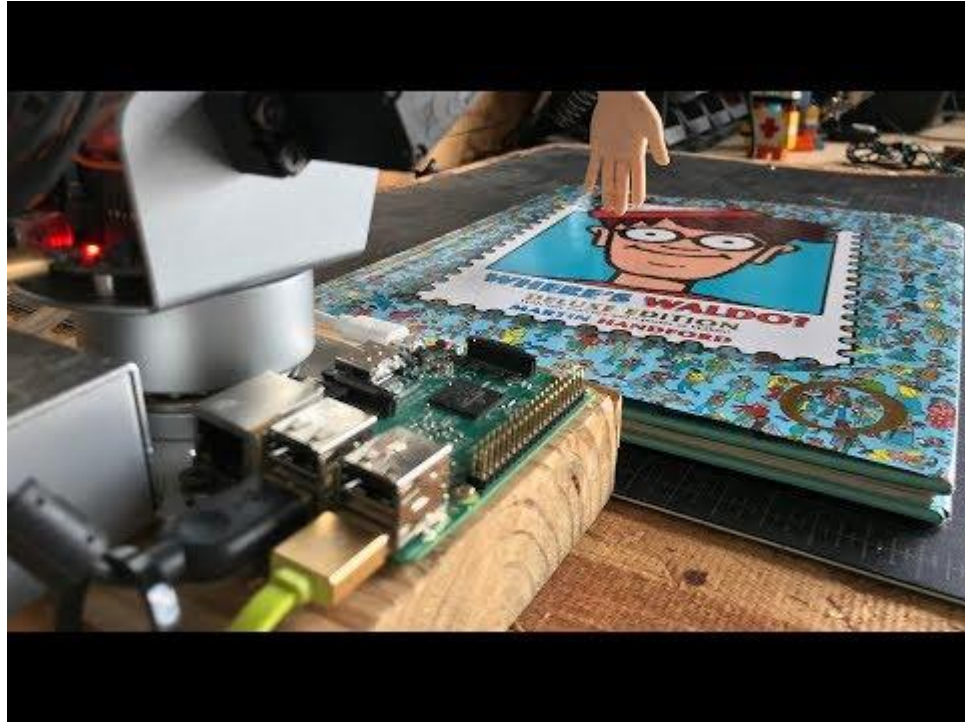
# Introduction to Machine Learning/Deep Learning (Cont.)

## Reinforcement Learning



# Introduction to Machine Learning/Deep Learning (Cont.)

## Convolution Neural Network



# Basic Pandas

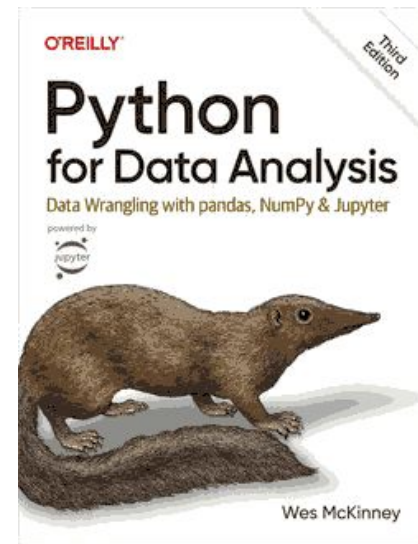
Pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool,

built on top of the Python programming language.

	0	1	2	3	4	
Column Label/ Header	Name	Age	Marks	Grade	Hobby	
Index Label						
0	S1	Joe	20	85.10	A	Swimming
1	S2	Nat	21	77.80	B	Reading
2	S3	Harry	19	91.54	A	Music
3	S4	Sam	20	88.78	A	Painting
4	S5	Monica	22	60.55	B	Dancing

Diagram illustrating the structure of a Pandas DataFrame:

- Column Index:** The top row (headers) is labeled with indices 0 to 4.
- Row Index:** The leftmost column (row labels) is labeled with indices 0 to 4.
- Column Label/ Header:** The headers are Name, Age, Marks, Grade, and Hobby.
- Index Label:** The row labels are S1, S2, S3, S4, and S5.
- Column:** A vertical slice of the data, e.g., the 'Marks' column.
- Row:** A horizontal slice of the data, e.g., the row for S4.
- Element/ Value/ Entry:** A single data point, e.g., 88.78.



# Basic Pandas (Cont.)

What Can You Do With **DataFrames Using Pandas**?

Pandas makes it simple to do many of the time consuming, repetitive tasks associated with working with data, including:

- Data cleansing
- Data fill
- Data normalization
- Merges and joins
- Data visualization
- Statistical analysis
- Data inspection
- Loading and saving data
- And much more

The diagram illustrates a Pandas DataFrame with 7 rows and 5 columns. The columns are labeled 'Name', 'Team', 'Number', 'Position', and 'Age'. The rows are indexed 0 through 6. Annotations include: 'Columns' pointing to the header row; 'Rows' pointing to the row indices; and 'Data' pointing to the cell contents. Some cells are highlighted with pink boxes: 'Jonas Jerebko', '8.0', 'Boston Celtics' (in the 'Team' column for row 3), 'PG', 'NaN', and '27.0'.

	Name	Team	Number	Position	Age
0	Avery Bradley	Boston Celtics	0.0	PG	25.0
1	John Holland	Boston Celtics	30.0	SG	27.0
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN
6	Evan Turner	Boston Celtics	11.0	SG	27.0

# Part 1: Pandas Code

[https://colab.research.google.com/github/kaopanboonyuen/GISTDA\\_TRAINING\\_2023/blob/main/code/C1-BasicPandas.ipynb](https://colab.research.google.com/github/kaopanboonyuen/GISTDA_TRAINING_2023/blob/main/code/C1-BasicPandas.ipynb)

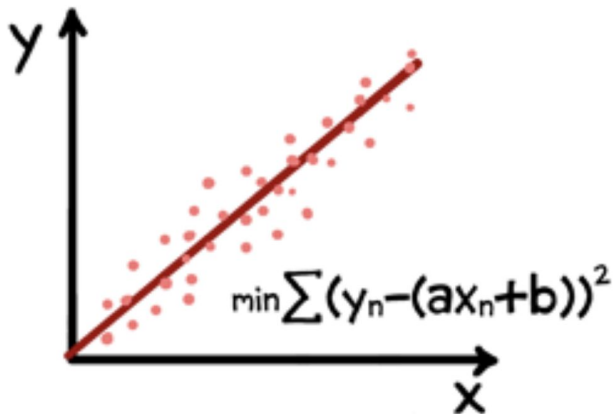




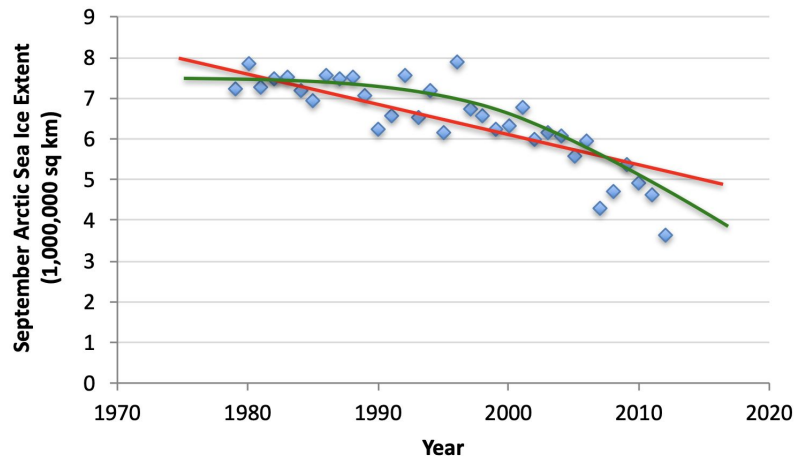
# Regression

Supervised Learning: Regression

## Linear Regression



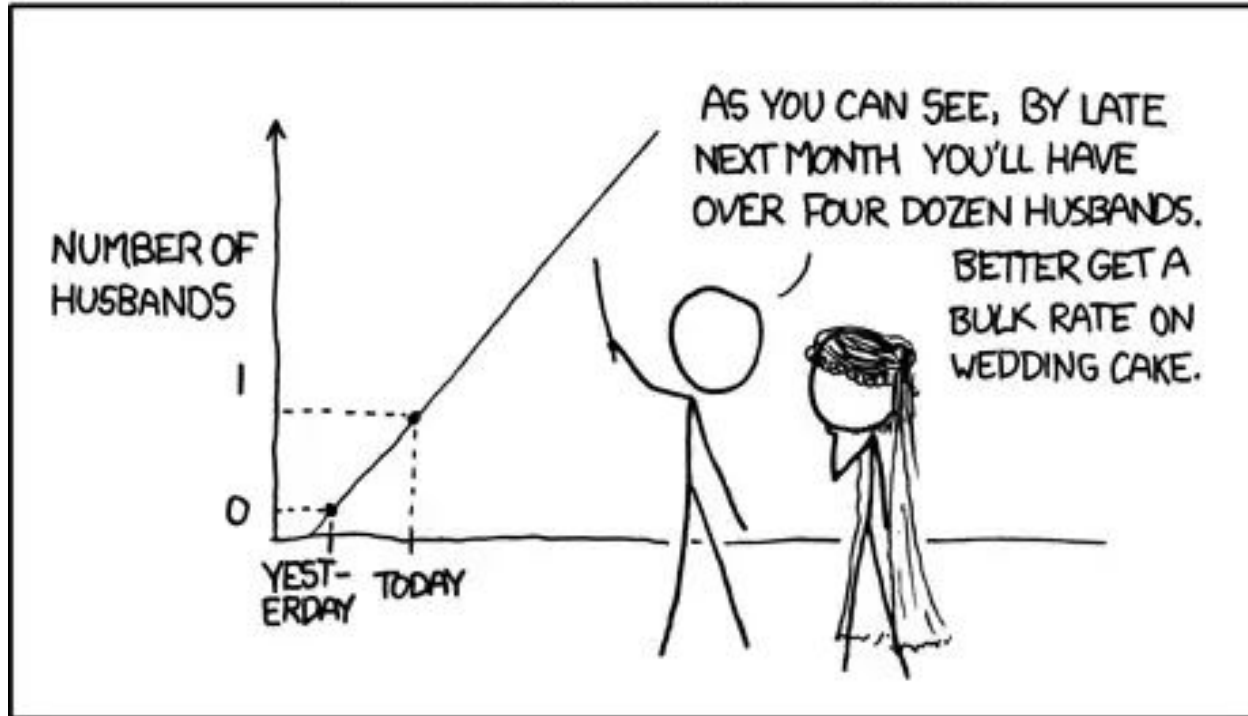
- Given  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$
- Learn a function  $f(x)$  to predict  $y$  given  $x$ 
  - $y$  is real-valued == regression





## Regression (Cont.)

MY HOBBY: EXTRAPOLATING



# Regression (Cont.)

## Types of Linear Regression

Linear regression can be further divided into two types of the algorithm:

### **Simple Linear Regression:**

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

### **Multiple Linear regression:**

If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

## Part 2: Regression Code

[https://colab.research.google.com/github/kaopanboonyuen/GISTDA\\_TRAINING\\_2023/blob/main/code/C2-Regression.ipynb](https://colab.research.google.com/github/kaopanboonyuen/GISTDA_TRAINING_2023/blob/main/code/C2-Regression.ipynb)



# SARIMAX

SARIMA stands for **Seasonal Autoregressive Integrated Moving Average** (quite a mouthful).

SARIMAX (Seasonal Autoregressive Integrated Moving Average **with eXogenous factors**)

It's very much like ARIMA but more powerful.

We can use statsmodels' implementation of SARIMA.

$$SARIMA \underbrace{(p, d, q)}_{non-seasonal} \underbrace{(P, D, Q)_m}_{seasonal}$$

# SARIMAX (cont.)

## Orders of the SARIMA model

A SARIMA model can be tuned with two kinds of orders:

**(p,d,q) order**, which refers to the order of the time series. This order is also used in the ARIMA model (which does not consider seasonality);

**(P,D,Q,M) seasonal order**, which refers to the order of the seasonal component of the time series.

# SARIMAX (cont.)

(P,D,Q,M) Order

The (P,D,Q,M) Order refers to the seasonal component of the model for the Autoregressive parameters, differences, Moving Average parameters, and periodicity:

**D** indicates the integration order of the seasonal process (the number of transformation needed to make stationary the time series)

**P** indicates the Autoregressive order for the seasonal component

**Q** indicated the Moving Average order for the seasonal component

**M** indicates the periodicity, i.e. the number of periods in season, such as 12 for monthly data.

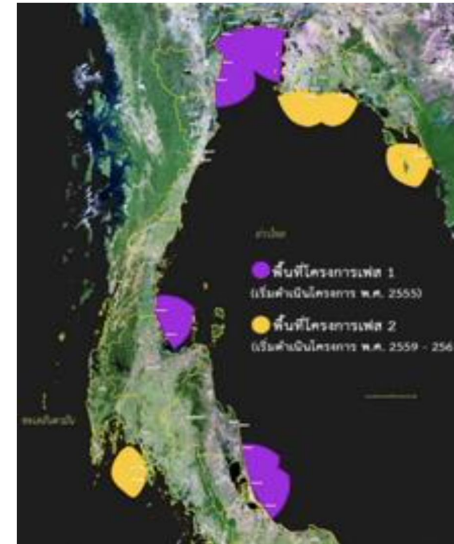
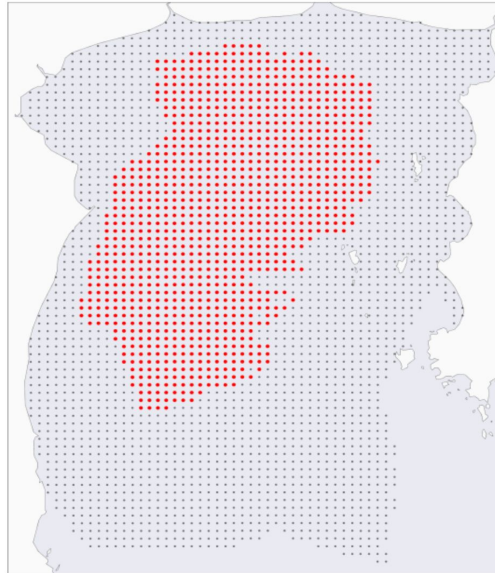
## Part 3: SARIMAX Code and GISTDA Ocean Data Exercise

[https://colab.research.google.com/github/kaopanboonyuen/GISTDA\\_TRAINING\\_2023/blob/main/code/C3-SARIMAX.ipynb](https://colab.research.google.com/github/kaopanboonyuen/GISTDA_TRAINING_2023/blob/main/code/C3-SARIMAX.ipynb)



# Exercise with GISTDA Ocean Current Data Set

[https://colab.research.google.com/github/kaopanboonyuen/GISTDA\\_TRAINING\\_2023/blob/main/code/C3-SARIMAX.ipynb](https://colab.research.google.com/github/kaopanboonyuen/GISTDA_TRAINING_2023/blob/main/code/C3-SARIMAX.ipynb)

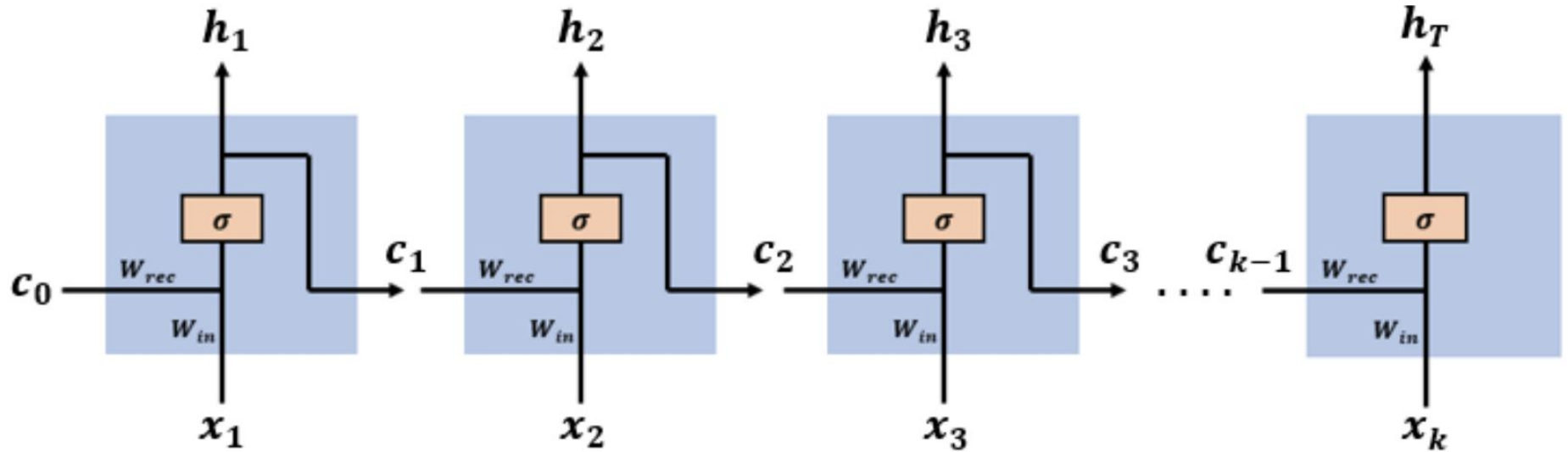


HF radar stations' area for both two phases (from <https://www.gistda.or.th>).



# illustrated Guide to LSTM and GRU

RNN ( Recurrent Neural Networks)



# illustrated Guide to LSTM and GRU (Cont.)

RNN takes input as time series (sequence of words ), we can say RNN acts like a memory that remembers the sequence.

the **LSTM (Long -short-term memory)** and **GRU (Gated Recurrent Unit)** have gates as an internal mechanism, which control what information to keep and what information to throw out.

By doing this LSTM, GRU networks solve the exploding and vanishing gradient problem.

# illustrated Guide to LSTM and GRU (Cont.)

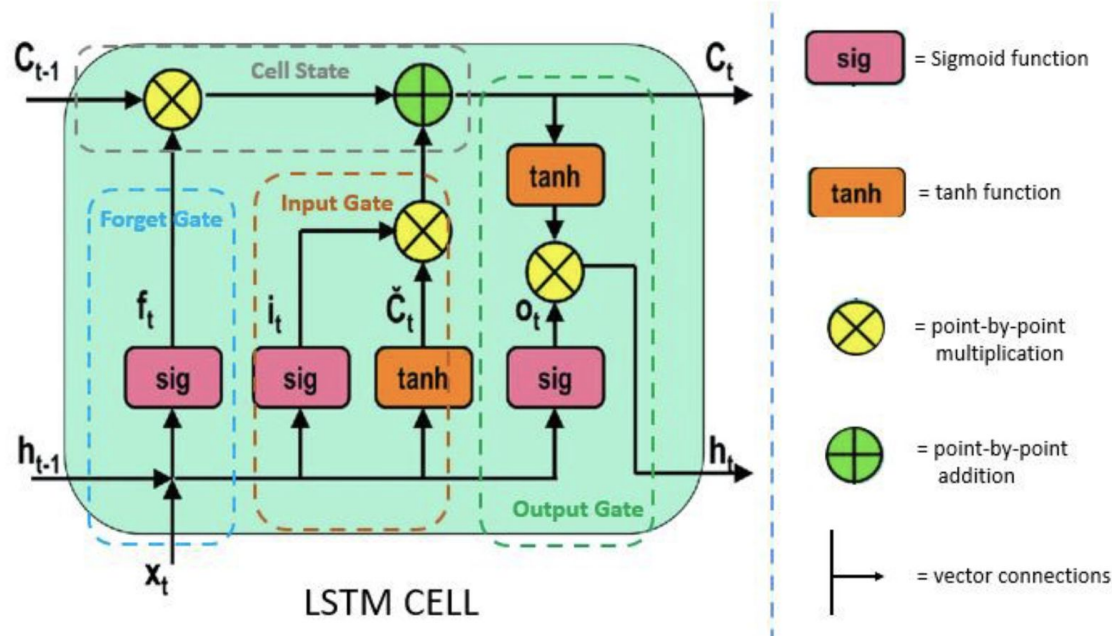
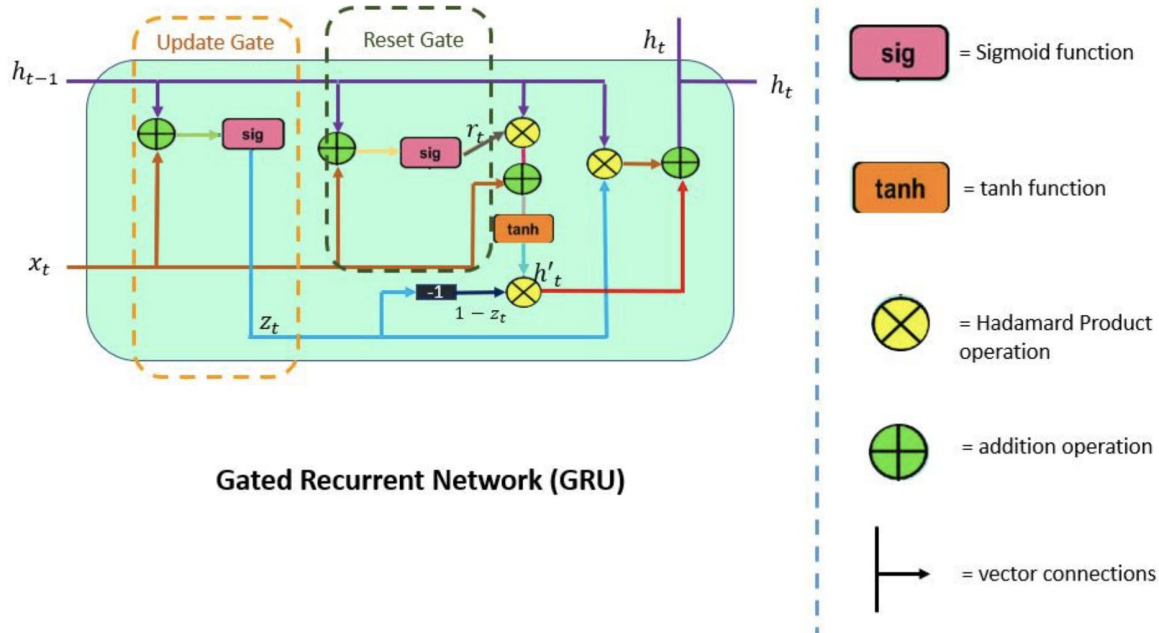


Image Source

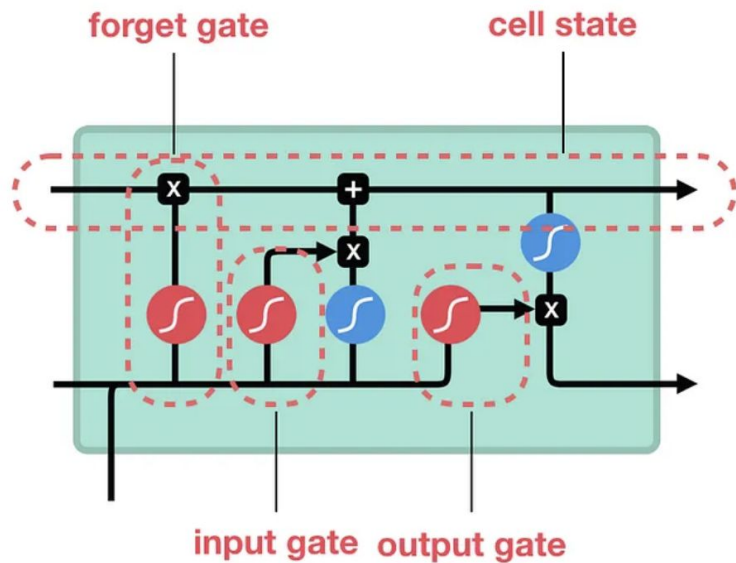
# illustrated Guide to LSTM and GRU (Cont.)

## GRU

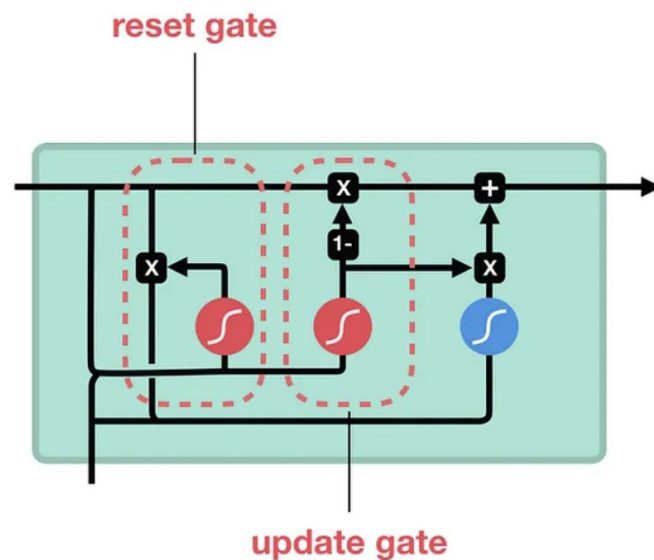


Gated Recurrent Network (GRU)

## LSTM



## GRU



sigmoid



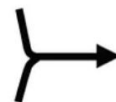
tanh



pointwise  
multiplication



pointwise  
addition



vector  
concatenation

## Part 4: Deep Learning (LSTM and GRU) Code

[https://colab.research.google.com/github/kaopanboonyuen/GISTDA\\_TRAINING\\_2023/blob/main/code/C4-LSTM-GRU.ipynb](https://colab.research.google.com/github/kaopanboonyuen/GISTDA_TRAINING_2023/blob/main/code/C4-LSTM-GRU.ipynb)



# Thank you

- All python codes/notebooks and lecture slide will be store through this github link [https://github.com/kaopanboonyuen/GISTDA\\_TRAINING\\_2023](https://github.com/kaopanboonyuen/GISTDA_TRAINING_2023)